Assignment 1:
Due: March 15, 2022, 11 PM

Overview: In this project you will write a spellchecker to spellcheck a file

Implementation
Your program should provide the following broad capabilities:

1. Read a list of words from jlawler-wordlist.txt (one word per line), and store those words in a PositionalList.
2. Read in a text file consisting of a series of English words (and possibly punctuation).
    - The file may be multiple lines long.
    - Use useDelimiter("[^a-zA-Z]+") when instantiating a scanner to eliminate punctuation
3. Spellcheck each word in the file. If it is a *misspelled* word, print it (lowercase), a dash, and then a list of possible substitutions. The misspelled words should be printed in the order in which they appear in the file: the substitutions should be printed in alphabetical order, in a comma-delimited list

    ```
    onec - donec, nec, once, one, ones
    upoon - poon, spoon, upon
    tiime - time
    ```
4. Display statistics, labeling the following (see the Sample output below)
    1. Number of words spellchecked (e.g., 4000)
    2. Percentage of words misspelled (e.g., 1.8%)
    3. Average number of suggestions made per word (e.g., 4.3%)
    4. Numbers of errors 1-4 that were encountered

Misspelled Words
A misspelled word will be caused by exactly one of the errors listed here. Do not check for compound errors, e.g., tery => tarry (replaced a with e and deleted an r)

1. the swapping of adjacent letters
2. the inadvertent insertion of an extra letter somewhere in the word,
3. the inadvertent deletion of a single letter from the word
4. replacing a single character with another

Requirements:
- You should identify the major units of functionality in this assignment, and use methods to encapsulate them. All methods must be commented using Javadoc and should be rigorously tested.
- Your code must be stored in a GitHub repo.
- Use net.datastructures implementations of any data structures required for this assignment.
- The usual coding standards apply. This project is to be done on your own: this is not a group project.
- Your output should look like mine (labels, number of digits after the decimal, spacing, etc.)

Deliverables:
Your entire Visual Studio Code project, zipped
an unzipped screenshot of your output, uploaded separately from the zipped project
in the submission comments, a link to your GitHub repo

Sample Output:

This is based on the text file shown at right.

```
Onec upoon q tiime,
a looong tiem aog,
there lived na eil
oger, unedr aa brridge.
```

```
onec - donec, nec, once, one, ones
upoon - poon, spoon, upon
tiime - time
looong - No Suggestions
tiem - diem, item, teem, tem, them, tie, tied, tien, tier, ties, time
aog - ago, agog, bog, cog, dog, fog, gog, hog, jog, log, nog
eil - ail, eel, eile, ein, el, ell, evil, heil, il, mil, nil, oil, veil
oger - goer, oder, oer, ogler, ogre, over, oyer, roger
unedr - under, uneder
aa - a, ab, aba, ac, ad, aga, ah, aha, al, ala, am, ama, an, ana, ar,
ara, as, at, au, ax, ay, ba, baa, ca, da, ea, fa, ga, ha, ja, ka, la, ma,
na, pa, ra, sa, ta, va, wa, za
brridge - bridge

# of words spellchecked: 16
% of words misspelled: 68.8
Average # of suggestions / misspelled word: 8.7
Swaps: 7
Insertions: 12
Deletions: 17
Replacements: 60
```